# An Application of Gradient-Like Dynamics to Neural Networks

James W. Howse Chaouki T. Abdallah
Gregory L. Heileman
Department of Electrical and Computer Engineering
University of New Mexico, Albuquerque, NM 87131
Michael Georgiopoulos
Department of Electrical and Computer Engineering
University of Central Florida, Orlando, FL 32816

**Abstract**

This paper reviews a formalism that enables the dynamics of a broad class of neural networks to be understood. This formalism is then applied to a specific network and the predicted and simulated behavior of the system are compared. A number of previous works have analyzed the Lyapunov stability of neural network models. This type of analysis shows that the excursion of the solutions from a stable point is bounded. The purpose of this work is to review and then utilize a model of the dynamics that also describes the phase space behavior and structural stability of the system. This is achieved by writing the general equations of the neural network dynamics as a gradient-like system. In this paper it is demonstrated that a network with additive activation dynamics and Hebbian weight update dynamics can be expressed as a gradient-like system. An example of a 3-layer network with feedback between adjacent layers is presented. It is shown that the process of weight learning is stable in this network when the learned weights are symmetric. Furthermore, the weight learning process is stable when the learned weights are asymmetric, provided that the activation is computed using only the symmetric part of the weights.

## Introduction

In studying the dynamics of unsupervised neural networks there are three critical issues which need to be analyzed. The first important issue is Lyapunov stability. It is important to establish conditions which guarantee that the node activities and connection weights converge to some equilibrium state of the network. The second important issue is the way in which the network stores information. This involves determining the nature of the equilibrium states in the network. The third important issue is the structural stability. This property determines whether a model can be made into a similarly functioning device, or whether the model can be simulated at a different level of precision (e.g. 8-bit vs. 16-bit). In order to do this, it is important to have some guarantee that small changes in the network parameters do not affect its general behavior.

Addressing all three of these concerns in a general neural network model can be quite difficult. In [2] the first of these problems is addressed by proving that a class of networks with a general equation for the node activation dynamics is Lyapunov stable when the weights are constant and symmetric. As shown in [6] many neural network models that do *not* include learning can be put in this general form. In [10], the aforementioned work is extended by using a similar equation for the node activation dynamics to prove the Lyapunov stability of networks with a number of different weight update rules. A different approach, which addresses all three of the issues discussed above, is taken in [12]. Specifically, some properties of a class of dynamical systems called gradient-like systems are derived and then used to explain some of the dynamics of the Hopfield network. We recently proposed a formalism [8] which extends the results in [12] by proving additional properties of gradient-like systems as well as allowing the incorporation of weight update in the gradient-like system formulation.

Gradient systems are a mathematically well studied class of dynamical systems. For such systems, results have been derived to address all three of the above concerns. We showed in [8] that most of the desirable properties of gradient systems are possessed by the more general class of gradient-like systems. We also demonstrated that many existing neural network models can be formulated as gradient-like systems. By contrast, few neural networks can be written as gradient systems. This formalism allows any dynamical system which can be cast as a gradient-like system to be analyzed with respect to its Lyapunov stability, phase space behavior, and structural stability.

This research was supported by a grant from Boeing Computer Services under Contract W-300445.

Lyapunov stability is used to determine whether most trajectories move toward or away from a given equilibrium. If an equilibrium state is Lyapunov stable, then any trajectory started in a given neighborhood of the equilibrium must have a bounded excursion from that equilibrium. Phase space behavior on the other hand, is used to evaluate the specific structure of the equilibria. The phase space is the space consisting of all state variables, and the collection of paths that the system state traverses in this space is called the phase space behavior of the system. It shows, for instance, whether the equilibrium state is a point, a periodic cycle, or some more complex behavior. Finally, structural stability is used to demonstrate whether small changes in system parameters change the qualitative system behavior. For example, in a structurally stable system the position of the equilibrium states in phase space remain similar under small variations of the system parameters.

In the next section we will give a brief review of the formalism developed in [8] and show that it applies to networks that have additive activation dynamics and Hebbian weight update dynamics. We will then present simulation results from a 3-layer feedback network which demonstrate the salient points of our theory.

## Review of Gradient-Like Dynamics

A gradient-like system is one in which the time derivative of the states $\dot{\boldsymbol{u}}$ is equal to the product of the gradient of a scalar function $V(\boldsymbol{u})$ and a symmetric positive definite matrix $\boldsymbol{P}(\boldsymbol{u})$. These dynamics are described by the equation

$$\dot{\boldsymbol{u}} = -\boldsymbol{P}(\boldsymbol{u})\left[\nabla_{\boldsymbol{u}} V(\boldsymbol{u})\right]. \tag{1}$$

The function $V(\boldsymbol{u})$ is a scalar function referred to as the gradient potential function. It is a mapping of the form $V : \mathcal{U} \to \mathbb{R}$, where $\mathcal{U} \subset \mathbb{R}^n$ is an open set, which is required to be twice continuously differentiable. The matrix $\boldsymbol{P}(\boldsymbol{u})$ must be symmetric and positive definite (i.e. $\boldsymbol{y}^T \boldsymbol{P}(\boldsymbol{x})\boldsymbol{y} > 0 \ \forall \ \boldsymbol{y} \neq \boldsymbol{0}$) for all values of $\boldsymbol{u}$.

Conceptually the function $V(\boldsymbol{u})$ defines a surface in the phase space of the system. All of the trajectories of the network must move along this surface. The matrix $\boldsymbol{P}(\boldsymbol{u})$ specifies the "laws of motion" that the trajectories must obey in moving along the surface defined by $V(\boldsymbol{u})$. Since $\boldsymbol{P}(\boldsymbol{u})$ is positive definite for all values of $\boldsymbol{u}$, the trajectories always move downhill along $V(\boldsymbol{u})$ (i.e. toward smaller values of $V(\boldsymbol{u})$). If a trajectory reaches a point where the slope of $V(\boldsymbol{u})$ is zero in *any* direction, then the trajectory remains at that point thereafter.

This intuition was formalized in [8] through a series of proofs which characterize the behavior of gradient-like systems. It was shown that every isolated local minima of $V(\boldsymbol{u})$ is an asymptotically stable equilibrium point of the network. This does *not* guarantee that every trajectory will converge to an equilibrium point. In order for that to occur the set

$$\mathcal{N}_c = \{\boldsymbol{u} \in \mathbb{R}^n : V(\boldsymbol{u}) \leq c\} \tag{2}$$

must be compact (i.e. closed and bounded) for every $c \in \mathbb{R}$. This is guaranteed to be true if $V(\boldsymbol{u})$ is bounded below (i.e. $V(\boldsymbol{u}) \geq \delta \ \forall \ \boldsymbol{u} \in \mathbb{R}^n$), and radially unbounded (i.e. $V(\boldsymbol{u}) \to \infty$ as $||\boldsymbol{u}|| \to \infty$).

As the intuitive description of gradient-like dynamics implies, the phase space behavior of such systems is quite simple. Since the trajectories can only remain constant at the equilibrium points and must move toward smaller values of $V(\boldsymbol{u})$ at all other points, the only recurrent trajectories are the equilibria themselves. A *recurrent trajectory* is one that returns to within an arbitrarily small neighborhood of its starting point at some later time. Since almost all trajectories of a gradient-like system must move down hill along the surface defined by $V(\boldsymbol{u})$, almost all trajectories end up at a stable equilibrium point or go to infinity. The exception to this is those few trajectories which terminate at a saddle point. Likewise all trajectories must begin at an unstable equilibrium point or at infinity. Furthermore, in gradient-like systems only three types of equilibria are possible, stable points, unstable points, and saddle points. In the next section we will show how to formulate a specific neural network as a gradient-like system.

## Neural Network Formulation

It is often useful to employ neural network models in which the node activation dynamics are described by an additive equation, and weight update dynamics are given by the Hebb rule. Some of the properties of networks using these dynamics are presented in [1, 5, 9]. These choices of network dynamics can be shown to fit into the gradient-like dynamics formalism. Consider a neural network with $p$ nodes and $m$ weights. The activation of the $i$th node is given by $x_i$, and the value of the weight *to* the $i$th node, *from* the $j$th node is given by $c_{ij}$. Following the form in [10], additive activation dynamics are described by the differential equation

$$\left(\frac{1}{\epsilon_i}\right)\dot{x}_i = -\mathcal{A}_i x_i + I_i + \sum_{j=1}^{p} c_{ij} d_j(x_j) \quad i = 1, \ldots, p. \tag{3}$$

In this equation $1/\epsilon_i$ is a constant which determines the speed at which $x_i$ converges to its equilibrium value. The term $-\mathcal{A}_i x_i$ is a passive decay term which causes $x_i$ to go to zero if the remaining terms are zero. The constant $\mathcal{A}_i$ determines the rate of decay. The function $d_j(x_j)$ is the output function of the $j$th node, and the input to the $i$th node is $I_i$. In equation (3) the inputs $I_i$ and the connection weights $c_{ij}$ may both take positive or negative values. Again following the form in [10], the dynamics of the Hebbian weight update rule are

$$\dot{c}_{ij} = -\gamma_{ij} c_{ij} + \lambda_{ij} d_i(x_i) d_j(x_j) \quad i, j \in \{1, \ldots, p\}. \tag{4}$$

The term $-\gamma_{ij} c_{ij}$ is a passive decay term where $\gamma_{ij}$ is a constant which determines the decay rate. The constant $\lambda_{ij}$ determines the growth rate of the connection weight $c_{ij}$ if the nodes at both ends of the connection are active. The matrices containing all such constants are $\Gamma$ and $\Lambda$ respectively.

In order to instantiate additive activation dynamics and the Hebbian learning rule into the gradient-like system of equation (1), define a state vector $\boldsymbol{u}$ as

$$\boldsymbol{u} = [x_1,\, x_2,\, x_3,\, \ldots,\, x_p,\, c_{11},\, c_{12},\, c_{13},\, \ldots,\, c_{pp}]^T. \tag{5}$$

Now let the gradient potential function be given by

$$\begin{aligned} V(\boldsymbol{u}) = -\frac{1}{2} \boldsymbol{d}(\boldsymbol{x})^T \boldsymbol{C} \boldsymbol{d}(\boldsymbol{x}) + \sum_{k=1}^{p} \int_0^{x_k} d_k'(\zeta_k)\,(\mathcal{A}_i \zeta_k - I_i)\; d\zeta_k \\ + \frac{1}{4} \boldsymbol{1}^T \left[ \boldsymbol{\Gamma} \circ \boldsymbol{\Lambda}^{-1} \circ \boldsymbol{C} \circ \boldsymbol{C} \right] \boldsymbol{1}. \end{aligned} \tag{6}$$

In equation (6) note that $\boldsymbol{1}$ is a $p$ dimensional vector whose elements are all 1. Also the operation $\circ$ denotes the *Schur* product which is defined as $[\boldsymbol{A} \circ \boldsymbol{B}]_{ij} = a_{ij} b_{ij}$. Since $V(\boldsymbol{u})$ must be twice continuously differentiable, the same requirement must hold for the output functions $d_i(x_i)$. Choose the matrix $\boldsymbol{P}(\boldsymbol{u})$ to be

$$\begin{aligned} \boldsymbol{P}(\boldsymbol{u}) = \Delta \Bigg[ \frac{\epsilon_1}{d_1'(x_1)},\, \ldots,\, \frac{\epsilon_p}{d_p'(x_p)}, \\ 2\lambda_{11},\, 2\lambda_{12},\, 2\lambda_{13},\, \ldots,\, 2\lambda_{pp} \Bigg]. \end{aligned} \tag{7}$$

The notation $\Delta[h_{11},\, h_{22},\, \ldots,\, h_{qq}]$ will be used to denote a $(q \times q)$ diagonal matrix with the listed elements along the diagonal. In order for $\boldsymbol{P}(\boldsymbol{u})$ to be positive definite, the constants $\epsilon_i$ and $\lambda_{ij}$ must be strictly positive numbers, and the output functions $d_i(x_i)$ must be monotonically increasing (i.e. $d_i'(x_i) > 0$). From equation (6) it is apparent that the gradient $\nabla_{\boldsymbol{u}} V(\boldsymbol{u})$ is

$$\nabla_{\boldsymbol{u}} V(\boldsymbol{u}) = \begin{pmatrix} d_1'(x_1) \left[ \mathcal{A}_1 x_1 - I_1 - \sum_{j=1}^{p} \frac{1}{2} (c_{1j} + c_{j1})\, d_j(x_j) \right] \\ \vdots \\ d_p'(x_p) \left[ \mathcal{A}_p x_p - I_p - \sum_{j=1}^{p} \frac{1}{2} (c_{pj} + c_{jp})\, d_j(x_j) \right] \\ \frac{1}{2} \frac{\gamma_{11}}{\lambda_{11}} c_{11} - \frac{1}{2} d_1(x_1) d_1(x_1) \\ \vdots \\ \frac{1}{2} \frac{\gamma_{pp}}{\lambda_{pp}} c_{pp} - \frac{1}{2} d_p(x_p) d_p(x_p) \end{pmatrix}. \tag{8}$$

It can be seen from equation (8) that there are two classes of networks whose gradient potential function is given by equation (6) which have gradient-like dynamics. The first class are those systems in which the weight matrix $\boldsymbol{C}$ learned by the Hebbian rule is symmetric. This will occur if the matrices $\boldsymbol{\Gamma}$ and $\boldsymbol{\Lambda}$ are symmetric, and the initial conditions for $c_{ij}$ and $c_{ji}$ are the same. A reasonable physical interpretation of this situation is that there is a single bidirectional link between any two nodes, rather than two unidirectional ones. The second class are networks in which the learned weight matrix $\boldsymbol{C}$ is asymmetric, but only the symmetric part of the weight matrix is used to calculate the node activations $\boldsymbol{x}$. It is shown in [7] that this treatment can be extended to incorporate anti-Hebbian learning [3], higher order networks [4, 11], and multiplicative node activation dynamics [6].

## Network Example

In this section we will present simulation results for a 3-layer recurrent neural network. The simulations will be used to illustrate the way in which the various properties of gradient-like systems appear in the dynamical behavior of the network. The network that will be simulated is illustrated in Figure 1. The output functions in all cases

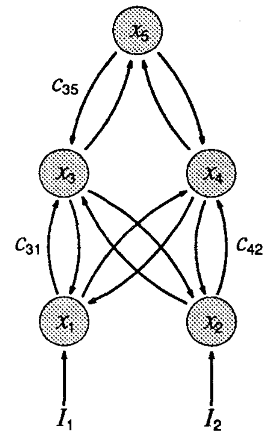


**Figure 1:** Configuration of example network

are $d_i(x_i) = \tanh(3\,x_i)$. First results will be shown for a network in which the learned connections are symmetric. For the simulation results which follow, the values of the parameters in equations (3) and (4) and the network inputs are

$$\begin{aligned} \mathcal{A}_i &= 1, & \epsilon_i &= 10, & \gamma_{ij} &= 2, \\ \lambda_{ij} &= 5, & I_1 &= 5, & I_2 &= 3, \end{aligned} \tag{9}$$

for $i = 1, \ldots, 5$ and $i, j \in \{1, \ldots, 5\}$. The fact that the learned connections weights are symmetric at all time values is best illustrated by a plot of complementary weight values (i.e. $c_{ij}$ and $c_{ji}$) versus time. The qualitative features of the dynamic behavior are best seen in a cross section of the phase space. Two representative plots of this type are shown in Figure 2. In the illustrated portion

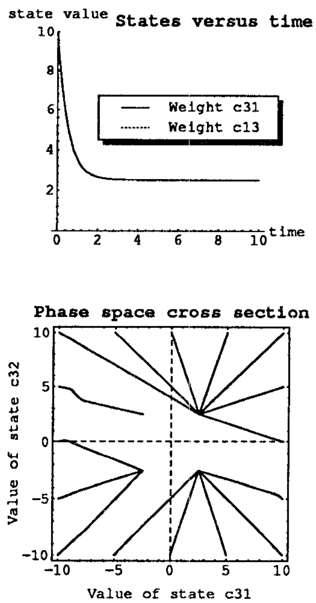


**Figure 2:** Phase and time plots when the learned connections are symmetric

of the phase space, this network converges to one of four stable equilibria depending on the initial conditions.

Next results will be shown for a network in which the learned connections are asymmetric but only the symmetric part of the weights is used to calculate the node activation values. The network parameters are the same as those in equation (9) except for

$$\begin{aligned} \gamma_{ij} &= 2, \text{ for } i < j, \qquad \gamma_{ij} = 4, \text{ for } i > j; \\ \lambda_{ij} &= 5, \text{ for } i < j, \qquad \lambda_{ij} = 3, \text{ for } i > j. \end{aligned} \tag{10}$$

Two representative plots of the complementary weights versus time and phase space cross section for this case are shown in Figure 3. Note that in this case there are only two stable equilibria in the same cross section of phase space as in the symmetric example. By allowing the learned weights to be asymmetric the number of equilibrium points as well as their location can be controlled. In addition, which trajectories approach a given equilibrium point can be modified in this way. Most importantly these results can be achieved with sacrificing the convergent properties of the network.

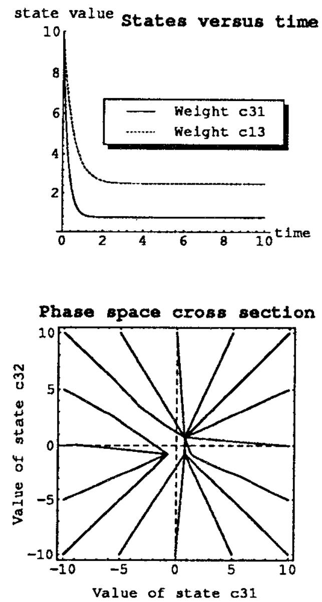


**Figure 3:** Phase and time plots when the learned connections are asymmetric

## Conclusion

The example in the last section shows that in principle it is possible to control the number, location, and region of attraction of the equilibrium points in a neural network without losing the convergence properties of the system. This can be accomplished by allowing the connections weights to be asymmetric while using only the symmetric part of the weights to calculate the node activation values. This is possible because the networks that have been considered have gradient-like dynamics to spite having asymmetric connections.

## References


[1] S.-I. Amari. Neural theory of association and concept-formation. *Biological Cybernetics*, 26(2):175–185, 1977.

[2] M.A. Cohen and S. Grossberg. Absolute stability of global pattern formation and parallel memory storage by competitive neural networks. *IEEE Transactions on Systems, Man and Cybernetics*, 13(5):815–825, 1983.

[3] P. Földiák. Forming sparse representaions by local anti-Hebbian learning. *Biological Cybernetics*, 64(2):165–170, 1990.

[4] C.L. Giles and T. Maxwell. Learning, invariance, and generalization in high-order neural networks. *Applied Optics*, 26(23):4972–4978, 1987.

[5] S. Grossberg. Pattern learning by functional–differential neural networks with arbitrary path weights. In S. Grossberg, editor, *Studies of Mind and Brain: Neural Principles of Learning, Perception, Development, Cognition, and Motor Control*, pages 159–193. D. Reidel Publishing Co., Dordrecht, Holland, 1982. volume 70 of *Boston Studies in the Philosophy of Science.*

[6] S. Grossberg. Nonlinear neural networks: Principles, mechanisms, and architectures. *Neural Networks*, 1(1):17–61, 1988.

[7] J.W. Howse, C.T. Abdallah, and G.L. Heileman. Gradient-like dynamics in neural networks. Technical Report EECE93–001, University of New Mexico, January 1993.

[8] J.W. Howse, C.T. Abdallah, G.L. Heileman, and M. Georgiopoulos. A gradient-like formulation of neural network dynamics. Submitted to *Neural Networks.*

[9] B. Kosko. Adaptive bidirectional associative memories. *Applied Optics*, 26(23):4947–4960, 1987.

[10] B. Kosko. *Neural networks and fuzzy systems: A dynamical systems approach to machine intelligence.* Prentice Hall, Inc., Englewood Cliffs, NJ, 1992.

[11] D. Psaltis, C.H. Park, and J. Hong. Higher order associative memories and their optical implementations. *Neural Networks*, 1(2):149–163, 1988.

[12] F.M.A. Salam, Y. Wang, and M.R. Choi. On the analysis of dynamic feedback neural nets. *IEEE Transactions on Circuits and Systems*, 38(2):196–201, 1991.